Effects of Forward Error Correction (FEC) on SURAN Protocol

SRNTN 48

February 1987

Prepared by: Neil Gower
Collins Defense Communications
Rockwell International

Prepared for: Dr. Dennis Perry
Defense Advanced Research Projects Agency
1400 Wilson Boulevard
Arlington, Virginia 22209-2308

Mr. Paul Sass
US Army CECOM
Attention: AMSEL-COM-AF-2
Fort Monmouth, New Jersey 07703-5202

Sponsored by: Defense Advanced Research Projects Age. .y (DoD)
Information Sciences and Technologies Office
Survivable Radio Networks (SURAN) Program
ARPA Order No. 4747/8
Issued by E. Clark under contract MDA903-85-C-0205

The views and conclusions contained in the document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of the Defense Advanced Project Agency or the U.S. Government.

SECURITY CLASSIFICATION OF THIS PAGE

ADA 182632

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188 Exp. Date: Jun 30, 1986

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY DD Form 254 dated 4 January 1986	3. DISTRIBUTION/AVAILABILITY OF REPORT APPROVED FOR PUBLIC RELEASE DISTRIBUTION UNLIMITED
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE CG-45	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) SURAN-205-12	5. MONITORING ORGANIZATION REPORT NUMBER(S) SRNTN 48
6a. NAME OF PERFORMING ORGANIZATION Rockwell International Collins Defense Communications 6b. OFFICE SYMBOL (If applicable) M/S 460-340	7a. NAME OF MONITORING ORGANIZATION Defense Advanced Research Projects Agency Information Sciences and Technologies Office
6c. ADDRESS (City, State, and ZIP Code) 3200 E. Renner Road Richardson, Texas 75081-6209	7b. ADDRESS (City, State, and ZIP Code) 1400 Wilson Boulevard Arlington, Virginia 22209-2308
8a. NAME OF FUNDING/SPONSORING ORGANIZATION DARPA 8b. OFFICE SYMBOL (If applicable) ISTO	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER Contract No. MDA-903-85-C-0205
8c. ADDRESS (City, State, and ZIP Code) 1400 Wilson Boulevard Arlington, Virginia 22209-2308	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. 62708E PROGRAM CODE 5T10 TASK NO. 4747/8 WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification)

Effects of Forward Error Correction (FEC) on SURAN Protocol

12. PERSONAL AUTHOR(S)

Neil Gower

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Technical Note	FROM 1985 TO 1987	1987 February	14

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	
17	02	.1	Packet Radio, Forward Error Correction, Protocol

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

This paper discusses effects of using forward error correction (FEC) in the Low-Cost Packet Radio (LPR) on the Survivable, Adaptive Networks (SURAN) protocol operating in the LPR.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION Unclassified
22a. NAME OF RESPONSIBLE INDIVIDUAL	22b. TELEPHONE (Include Area Code) / 22c. OFFICE SYMBOL

DD FORM 1473, 84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE

Effects of Forward Error Correction (FEC) on SURAN Protocol

INTRODUCTION:

As we look at the FEC effects, we note that other than the obvious benefit of being able to reclaim otherwise mangled packets there are some implications to the throughput and delay of the network at the link level. Certainly, the fact that otherwise demolished packets are now valuable will increase throughput and decrease delay. The fact that encoded packets require more time to transmit decreases throughput and increases delay. And certainly the processes of encoding and decoding take time. For the purposes of this paper, we will assume that the decision to use FEC has already been made manually or according to some algorithm and deal with the impacts on the SURAP algorithms and suggest possible approaches to handling them.

Time is relative and from past experience and intuition it is easy to see that time should be measured relative to the most precious resource - radio channel. The encoding process is rather fast compared to the time required to transmit a packet so that other than a pipeline effect, there is little effect on the throughput of the network - primarily delay. This does not include the fact that the simple act of transmitting an encoded packet costs channel time due to the increase in number of bits (symbols) transmitted per packet. The process of decoding on the other hand can vary from requiring much less time than the transmission, when no errors are incurred, to requiring much more time than the transmission, when many errors are incurred. If sufficient errors are present in the packet to be decoded, the decoder will in essence NEVER finish. For this reason, a maximum time to allow the decoder to try to resurrect a packet, or time-out, has been incorporated into the LPR.

With this perspective on FEC and the premise that FEC will be used on a regular basis for the SURAN protocols, several issues must be resolved. How can the time to attempt to decode be kept to an acceptable level and still optimize the usefulness of the FEC? Is the time-out the same for all types of packets or are there some simple rules that can be applied to allow tne optimum use of the FEC? What is the effect of the decode time on the pacing algorithms that are being implemented in the protocols?

USE of FEC:

After studying the frame nature of the forwarding process with the decoding times included, it becomes apparent that the decode process can be thought of as an extension to the processing in the packet radio. For any given packet, the transmit, receive, encoding, decoding and protocol-processing are all in series and this reinforces the notion of the pipeline nature of the processes involved. This concept weighs heavily in all the questions mentioned before. It would be best if ALL other processes combined required no more time than that of radio transmission, because this would provide the least delay and the maximum throughput. However, the next best is that the processes be parallel pipelines (see figure I), because the same maximum throughput would be maintained in the case that the node is handling cross traffic. If one continues with the concept of pipelining, the processes might be considered parallel pipelines processing parallel packets (to/from different PRs). In order to attain efficient parallel processing (pipelines) of packets, no process in the series should require more time than the most precious process, that of radio transmission. If the decoder pipeline delay is allowed to grow past that of the radio transmission pipeline delay, then it becomes the pacing item.

Using the logic of parallel pipelines one could suggest using a decoder time-out approximately equal to the packet's transmission time. The decoder operates at a computation rate of 1.28mcmp/s (mega-computations/sec). The decoder processes two symbols per computation cycle, if there are no errors in the encoded packet. Therefore a packet transmission time at the rate of 400ksps (kilo-symbols per second) would translate to 6.4 times the time to decode an error-free packet. With the data currently available to us, we do not know exactly what degree of errors (Bit Error Rate - BER) this would resolve. It is known that the BER value which corresponds to this decoding time will vary with the type of errors incurred (Gaussian, pulse, etc.). As a statistical choice for the purposes of selecting a value for the decoder time-out, we have been told to allow ten times the error-free decode time to decode a "worst-case" packet. This translates to more than 1 and 1/2 times the packet transmission time.

One approach is to allow a longer (than packet transmit time) time-out, as long as there is not a "building" backlog of packets to beencoded/decoded. In this way the decoder would be allowed a longer time to work on packets when it is not the limiting factor in the throughput chain. When the backlog builds, then the decoder is allowed only the time it took to receive the packet. As the backlog is worked off, the time-out is allowed to increase again. Some action similar to this will be required to avoid totally missing packets due to a shortage of packet buffers. Remember that even though a longer decode time may provide a time for other PRs to transmit, the PR must decode these packets, too.

FIGURE I

```
(PKT 1) ----->     (PKT 2) =====>     (PKT 3) *****>     (PKT 4) #####>

  NOTE: MULTIPLE OUTSTANDING PACKETS ARE TO DIFFERENT "NEXT" PRs.
```

MINIMUM DELAY TIMING - ALL PROCESSING TIME < TRANSMIT/RECEIVE TIME

```
RCV
  ----->=====>            -ACK->=ACK=>*****>#####>            *ACK*>#ACK#>
DECODE
        -->   ==>               A->   A=>   **>   ##>               A*>
PROTOCOL
           -->   ==>               A->   A=>   **>   ##>               A*>
ENCODE/XMIT
              ----->=====>                        *****>#####>

  |<- MINIMUM PACKET HANDLING TIME ->|
```

PARALLEL PIPELINED PROCESSES - OPTIMUM USE OF CHANNEL

```
RCV
  ----->            =====>            *****>      -ACK->#####>      =ACK=>
DECODE
        ----->            =====>            *****>      -ACK->
PROTOCOL
              ----->            =====>            *****>      -ACK->
ENCODE/XMIT
                          ----->            =====>            *****>

  |<-------------------- PACKET HANDLING TIME -------------------->|
```

Along with this discussion there is the thought that the decoder might also be allowed to have a longer decode time available when the pacing would prevent any packets in the transmit queue from being transmitted for at least the amount of the longer decode time-out. This is to say that the transmit function would not require the service of the encoder.

On a network packet with a routing header and text, the shorter time-out should still result in the decoding of the header. This will allow the disposition of many packets in the classes of PASSIVE_ACKs and PACKETS_NOT_FOR_THIS_PR, even if only the header is correctly decoded in the time allotted.

There are two packets which are of little or no value if only partially decoded. These are the PROP and the ACTIVE_ACK. The PROP in particular should have every opportunity to be received and decoded to maintain network operations. The ACTIVE_ACK can be solicited again, but due to its short packet length it would be well to expend the extra time to be assured of its decoding. Since the only information the LPROS has about the received packet before decode include the preamble fields and the DMA receive count (encoded packet length received), it is suggested that a bit in the preamble be used to indicate that these packets are types which contain only one checksum and therefore should be allowed a longer decode time, if necessary. Packets which contain two checksums (i.e. network packets with header) can be allowed to utilize whatever the current time-out is (short or normal).

A refinement to the idea would be to define the bit in the preamble as BROADCAST mode, as others have suggested for other reasons. The bit would then be set for PROPs or other future broadcast mode packets and the LPROS could discern that the packet would not be retransmitted and allow more time to decode. This would not accommodate the ACTIVE_ACK, which could be recognized by its short, header-only length, which the protocol could provide to the LPROS in order to allow it to be independent of revisions to the header.

The concept may be further enhanced by having the IOP "hold onto" the encoded packet, if the decode times out, until the protocol can look at the header to see if it wants to "go around again" with a longer time-out. While the total time to decode this packet will certainly be longer than desirable, it is expected that the number of these packets would be small compared to the ones that are serviced with the current time-out. This technique would be useful when the time-out is normal as well as shortened to reclaim packets directed to this PR, which would otherwise have to be retransmitted. Even with the "normal" time-out, there will be some small percentage of packets which will not fully decode. Significant problems are associated with this suggestion, one of which is the fact that the time lost in overhead functions of decode are incurred twice and could be relatively large. Non-trivial changes to the IOP software are also implied.

It is worth noting that the decode and encode processes share the same hardware and therefore are mutually exclusive. While the decode process is an "off-line" process, the encode process is an extension of the transmit. What this means is that transmissions will be held up if a long-running decode is in process, when transmission time occurs. When the FEC hardware becomes available, transmissions (encode) should take precedence. Since the encode process is much faster than the transmission process, when both a transmission and a decode are pending, the transmission should be started first and the decode of the received packet may be initiated to overlap the portion of the transmission after encode is complete. The current operating system (LPROS) implementation incorporates this concept.

There exists as a function of the design of the decoding hardware a minimum encoded packet length of 48 bytes in order to operate the decoder. For rates of 3/4 and 7/8, this may imply "stuffing" extra non-useful data in the packet at transmission. It has been suggested in the past and bears repeating that rather than send some garbage at 7/8 one should consider sending just good data at 3/4 or 1/2 to attain the minimum packet length, and rather than filler bits at 3/4 one should consider sending only real data at 1/2. As an example, ACTIVE_ACKs are only 12 words (24 bytes). If FEC is to be used for an ACTIVE_ACK always use rate 1/2, which will yield exactly 48 bytes of encoded symbols.

Another facet to the minimum encoded packet length is that the length limitation is actually the length of the packet that is passed to the decoder and includes the soft decision bits inserted at the receiver when this mode is invoked by the receiving PR. Since there is a soft decision bit for every received symbol, the actual minimum transmitted packet length for a packet which is to be received using soft decision is one-half that of one to be received using hard decision. Unfortunately, the selection of the soft decision decode mode is strictly a receive function. If some technique can be designed for the transmitting PR to know when soft decision processing will be used, then the minimum packet length could be decreased in those cases, reducing channel usage.

It is noted that the length of encoded packets must be on an integer byte boundary. This requires stuffing in many cases when rates 3/4 and especially 7/8 is used. Since the unencoded packet is in integer bytes and the encoded output must be in integer bytes, the 7/8 rate really implies a length that is an integer multiple of 7 bytes and the 3/4, an integer multiple of 3 bytes. Shorter packets will incur a more significant percentage of stuff bits. Consideration should be given to using a rate with more symbols, when significant stuffing will be required. That is, rate 1/2 will never require stuffing to integer byte boundary.

In order to quantify the effects of error-generating environments on the operation of the protocol using FEC in the network, tests could be run using an LPR to encode packets and apply varying degrees (BER) and types (Gaussian, pulse,etc.) of errors on them and then to decode them using a rather long time-out value. The errors would be generated and then de-interleaved before being superimposed on the data to be decoded. In this fashion the operation of the errors on the interleaved data can be simulated. Decoding time histograms would be kept for each type and degree of errors to gain a statistical probability of packets being decoded in varying error-generating environments. This data would then be used in setting the time-out value(s) for the protocol and evaluating the merit of the preceding proposals to enhance the decoding process.

PACING with FEC:

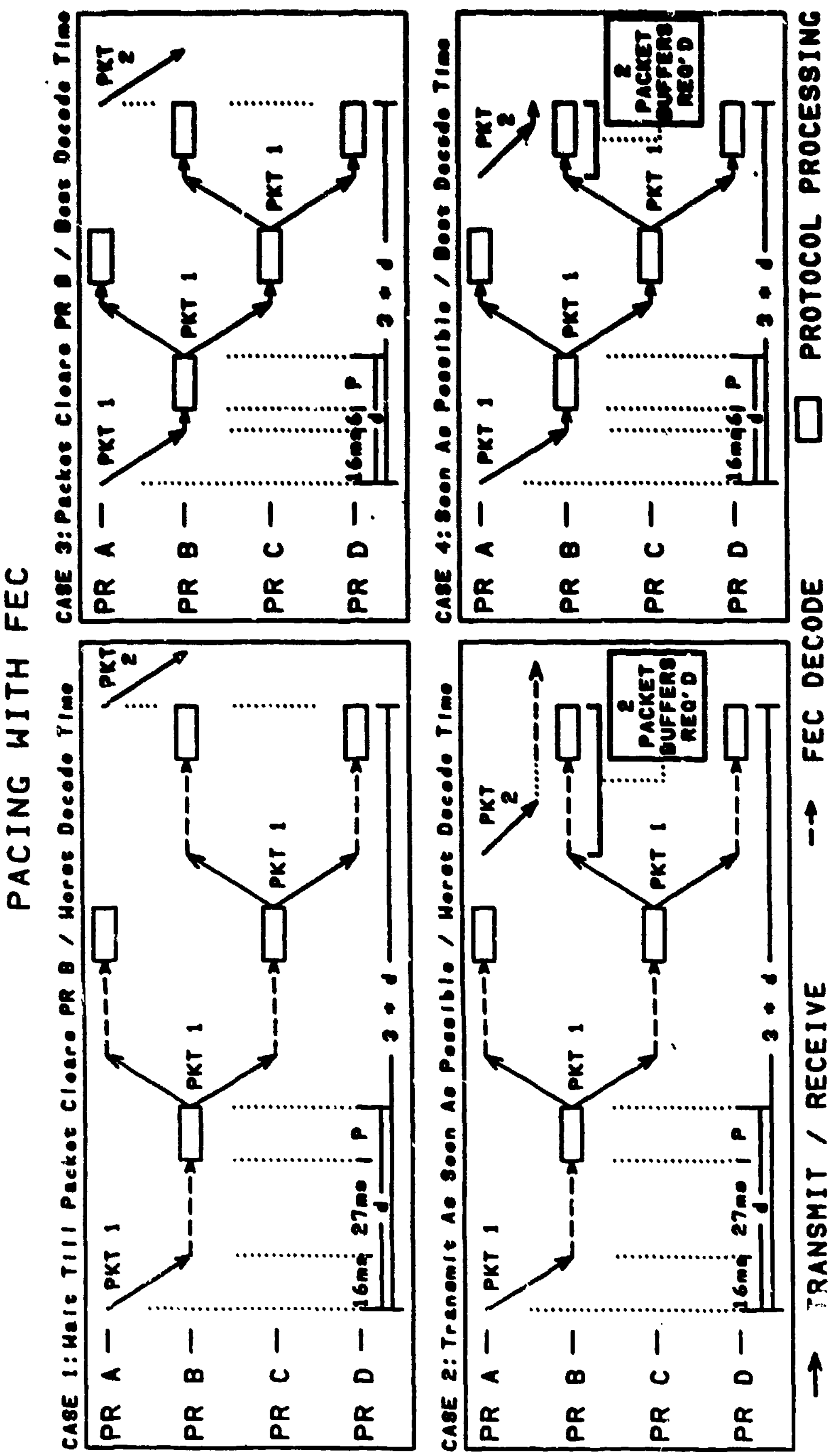
The LPR incorporates bit-by-bit code changing, which, in conjunction with FEC, will help to alleviate the collision problems due to hidden PRs. Pacing should still be used to avoid overlap with the next hop ACK to minimize the chance that either packet will have to be retransmitted, losing more time than waiting the full 3-times-the-delay period. Suppose (in the diagram below) that the transmission of the next packet (PKT 2) is not paced and conflicts with the transmission of the ACK of the previous packet (PKT 1) at the next PR (PR B). Even if PR B is able to receive one of the two packets, the flow is perturbed.

```
             PKT 2         PKT 1 ack
        O ----------> O <---------- O
       PR A          PR B          PR C
```

If PR B receives the ACK, PR A will have to retransmit the new packet. If the PR B receives the new packet instead, PR B will repeat the previous packet and wait for the ACK from PR C and the new packet will likely also be retransmitted by PR A, even though it is in the queue of the next PR. This example assumes that one of the packets is received by PR B. There is a real possibility that if the preambles overlap that neither will be received.

Figure II illustrates the timing of packet forwarding with packets encoded at rate = 1/2. Cases 1 and 2 illustrate the effect of two different forwarding delay policies, when the decode process requires a maximum time. Cases 3 and 4 illustrate the two policies when the decode process requires a minimum time. The encode time has been omitted for simplicity. The encode time is short compared to transmit and decode and is a real-time part of the transmit sequence. With the realization that the decode process is in series with all other processes, it is reasonable to extend the pacing algorithm to include it as part of the processing time. Cases 1 and 3 illustrate the policy of allowing enough time for the ACK to have been received, decoded and processed by the next PR. Cases 2 and 4 illustrate a policy of allowing only enough time that the ACK is received by the next PR. While it might seem to be desirable to pace packets at less than the multiple of approximately 3 times the measured delay to take advantage of the longer delay through a PR, this would require a PR to have more than one packet buffer available to a previous PR (Cases 2&4). While reducing the multiplier below three would possiblly result in a gain in throughput for the case of a string carrying traffic in one direction, a factor of less than three would not allow enough time for the forwarding of bi-directional streams of traffic. The figures assume that the decode time at each node is the same. While this may not be the true case, it is compatible with the assumption that each successive node will see the same processing and queuing delays (a premise which is at the root of the pacing algorithm).

FIGURE II


PACING WITH FEC
CASE 1: Wait Till Packet Clears PR B / Worst Decode Time
CASE 2: Transmit As Soon As Possible / Worst Decode Time
CASE 3: Packet Clears PR B / Best Decode Time
CASE 4: Soon As Possible / Best Decode Time
PR A
PR B
PR C
PR D
PKT 1
PKT 2
16ms
27ms
d
P
3 + d
2 PACKET BUFFERS REQ'D
TRANSMIT / RECEIVE
FEC DECODE
PROTOCOL PROCESSING

SUMMARY:

- It is recommended to use a decoder time-out somewhat longer than the packet receive time when there is no backlog of packets which must be serviced by the FEC function of the LPR.

- It is recommended to use a decode time-out equal to the time required for the packet to be received, when a backlog has developed.

- It is noted that many packets which have a separate header checksum, which are not fully decoded when the allotted time for decode elapses, are still useful as passive acks and for overheard packets which are not intended for this PR (currently implemented in SURAN protocols).

- It is possible to allow the LPROS to make decoding time-out decisions on the fly based on a bit in the preamble (set by the protocol of the transmitting PR) and possibly on the length (header only = ACTIVE_ACK). In this case, the bit being set would indicate that the packet will not be retransmitted (PROPs and ACTIVE_ACKs) and extra time may be allowed for decode.

- It is noted that it might be possible (though possibly difficult) to modify the IOP to allow the retention of the encoded packet buffer for packets which have been not fully decoded at the end of the allotted time, in order to decide if they should be run through again with a longer time-out.

- It is noted that packets which will be less than the minimum length after encoding at a higher rate, might be encoded at a more powerful, lower rate rather than "stuffing" useless bits. An example is that ACTIVE_ACKs would always use 1/2 rate if FEC is invoked.

- It is noted that if a technique were available for the transmitting PR to know that the receiving PR(s) would use soft decision, the minimum transmitted packet length could be cut in half.

- It is recommended to continue to use a value of 3 to multiply by the measured (and smoothed) delay through the next node for a pacing time.

APPENDIX:

PACKET TIME CALCULATIONS

FOR FULL LENGTH PACKET (192 WORDS):

1/2 RATE HARD-DECISION => 6144 TRANSMITTED SYMBOLS

1/2 RATE SOFT-DECISION => 6144 TRANSMITTED SYMBOLS + 6144 QUALITY BITS

DECODING TIME FOR FULL LENGTH 1/2 RATE PACKET TEXT

			HARD DECISION	SOFT DECISION
BEST CASE:				
COMP. TIME	= (6144 / 2) / 1.28 Mcmp	=	2.4 ms	2.4 ms
I/O TIME	= (6144 / 8) * 2 us	=	1.55ms	
I/O TIME	= ((6144 + 6144) / 8) * 2 us	=		3.1 ms
TOTAL DECODE TIME			3.95 ms	5.5 ms
WORST CASE:				
COMP. TIME	= 10 * (6144 / 2) / 1.28 Mcmp	=	24. ms	24. ms
I/O TIME	= (6144 / 8) * 2 us	=	1.55ms	
I/O TIME	= ((6144 + 6144) / 8) * 2 us	=		3.1 ms
TOTAL DECODE TIME			25.55 ms	27.1 ms

TIME TO TRANSMIT 1/2 RATE FULL LENGTH PACKET:

192 WORDS + CRC + WORD_0 + WORD_1 = 3136 BITS
@ 400kbps => 15.68 ms + PREAMBLE => 15.96 ms

SRNTN DISTRIBUTION LIST

Bolt Beranek and Newman, Inc. (2)
ATTN: Greg Lauer
10 Moulton Street
Cambridge, MA 02238

Defense Advanced Research (2)
Projects Agency/ISTO
ATTN: Dennis Perry
1400 Wilson Boulevard
Arlington, VA 22209

MIT Lincoln Laboratories
ATTN: Richard Ralston
Box 73, Room C-117
Lexington, MA 02173

Naval Warfare Systems Command
ATTN: Barry Hughes
Code 611
Washington, D.C. 20363-5100

Polytechnic Institute
ATTN: Bob Boorstyn
333 Jay Street
Brooklyn, NY 11201

SRI International (2)
ATTN: Janet Tornow
Office EJ131
333 Ravenswood Avenue
Menlo Park, CA 94025

SRI International
ATTN: Mike Frankel
Office EJ347
Menlo Park, CA 94025

UCLA
Computer Science Dept.
ATTN: Leonard Kleinrock
3732 BH
Los Angeles, CA 90024

USC-ISI
ATTN: Jon Postel
4676 Admiralty Way
Marina Del Rey, CA 90292-6695

Naval Research Lab
ATTN: J.E. Wieselthier
Code 7520
Washington, D.C. 20375

Paul Sass
US Army CECOM
ATTN: AMSEL-COM-AF-2
Fort Monmouth, NJ 07703

Hazeltine Corporation (2)
Building 1
ATTN: Jeff Markel
Cuba Hill Road
Green Lawn, NY 11740

National Security Agency
ATTN: S743, S. Spano
Fort George Meade, MD 20755

Naval Ocean Systems Center
ATTN: Dan Leonard
271 Catalina Boulevard
San Diego, CA 92152

Rockwell International (2)
Collins Defense Communications
ATTN: John Jubin M/S 460-340
3200 E. Renner Road
Richardson, TX 75081-6209

Horst Clausen
Electrical & Computer Engineering
University of Kansas
1013 Learned Hall
Lawrence, KS 66045-2228

Stanford University
Computer Systems Laboratory
ATTN: Fouad Tobagi
Stanford, CA 94305

University of Illinois
CSL
ATTN: Mike Pursley
1101 W. Springfield Road
Urbana, IL 61801

Rome Air Development Center
ATTN: Dan McAuliffe
DICEF-Section (DCLS)
Griffiss AFB, NY 13441

M/A-COM Government Systems
ATTN: Richard Binder
8619 Westwood Center Drive
Vienna, VA 22180

Defense Technical Information Center (2)
Cameron Center
Alexandria, Virginia 22314

[7 Nov 86]

Dr. Barry M. Leiner
RIACS
NASA Ames, M/S 230-5
Moffett Field, CA 94035